REMARKS/ARGUMENTS

The Office Action mailed July 21, 2006 has been carefully considered. Reconsideration in view of the following remarks is respectfully requested.

In the specification, the paragraphs 14, 16, 17 and 20 have been amended to correct minor editorial problems. No new matter has been added.

Claims 1, 18, and 23 have been amended to further particularly point out and distinctly claim subject matter regarded as the invention. Support for these changes may be found in the specification, drawings, and claims as originally filed. Applicants resepctfully submit, therefore, that no new matter has been added.

35 U.S.C. § 103 Rejection

Claims 1-39 were rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Block et al.[1] in view of Malloy et al.[2], among which claims 1, 18 and 23 are independent claims. Applicants respectfully submit that the claims are not anticipated or rendered obvious by the cited references alone or in combination. Applicants respectfully submit that block does not disclose or render obvious the limitation of a selection indicating internal metadata to be mapped to external metadata in an external system. In contrast, Claim 1 as amended, includes the following limitations.

[1] U.S. Publication No. US 2003/0037038
[2] U.S. Publication No. US 2004/0122844

A method for establishing a mapping between internal metadata and external metadata in a report design environment, the method comprising:

organizing internal metadata in a grid having rows and columns, wherein dimensional metadata from said internal metadata is placed in the grid as row headings and/or column headings;

receiving from a user a selection of a portion of said grid, said selection indicating internal metadata to be mapped to external metadata in an external system;

receiving from said user a definition of external metadata describing all data points within said selection; and

creating a mapping between said selected internal metadata and said defined external metadata.

(Amended claim 1) (Emphasis added)

Applicants respectfully submit that Block does not disclose this limitation. A thorough reading of Block makes clear that what is disclosed in Block is a method for transforming data streams into various formats within a system. As cited by the Examiner, Block discloses the following.

"In an exemplary embodiment of the invention, a data stream is captured, data in the captured stream are identified, and then the identified data are mapped to a file structure, a schema, or a taxonomy. In exemplary embodiments of the invention, the output data stream is a data stream to a display screen, a memory, a hard drive, a CD ROM drive, a floppy disk drive, or a printer. The output data stream can be conveyed through serial or parallel ports (including Universal Serial Bus or "USB", FireWire.TM.), via wireless interfaces, and so forth. In other exemplary embodiments of the invention, the identified data are mapped to an XBRL (eXtensible Business Reporting Language) taxonomy, a spreadsheet, a database, or a flat file.

In another exemplary embodiment of the present invention, a method for adding labels to data includes a) identifying data in an electronically represented file, b) selecting labels that correspond to text strings in the identified data, based on a list associating labels with text strings, and c) adding the selected labels into the electronically represented file to label the text strings and elements in the identified data associated with the text strings. The labels include information about the data and are defined in one or more taxonomies. In the event the list does not associate a label with the text string, a user can be prompted to select a label corresponding to a text string in the identified data. The association indicated by the user's selection, can then be added to the list associating labels with text strings. Preferably the labels are consistent with XML (extensible Markup Language), and also conform to an XBRL (extensible Business Reporting Language) specification. This embodiment can be implemented by a transformation

program that receives the electronically represented file from a target program. The transformation program a) performs the steps of identifying, selecting and adding, and b) is configured to appear to the target program as a printer driver. The transformation program can be independent and separate from the target program.

In accordance with another embodiment of the invention, a method is provided for importing at least a portion of an XBRL compliant data set into a non XBRL compliant target application. The method includes the steps of exporting data from the target program in an export file, a user associating entries in the export file with labels defined in one or more appropriate XBRL taxonomies, and forming an import file for import into the target program by replacing data in the export file at entries associated with specific labels, with data from the data set having corresponding labels. The associations made by the user are stored for later use, so that an import file can be automatically created by replacing data in a file having the same format as the originally exported file, based on the stored associations".

(Block, paragraphs 14 – 16)

These portions of Block make clear that what is described is the addition of internal metadata to internal data or to external data. Block does not disclose or suggest a correlation between internal metadata and external metadata.

Applicants maintain that Block fails to disclose or suggest the limitation of receiving a definition of external metadata describing all data points within said selection. As discussed, Block does not disclose or suggest the use of external metadata.

Furthermore, applicants respectfully submit that there is no teaching or suggestion to combine the cited references. Block teaches a method for transforming data streams into various formats within a system, while Malloy discloses a system for processing queries in a multi-dimensional OLAP system. Malloy dislcoses the following.

Disclosed is a method, system, and program for query processing. Metadata for a facts metadata object and one or more dimension metadata objects that are associated with the facts metadata object is stored. A view with columns for one or more measures in the facts metadata object and one or more attributes in the one or more dimension metadata objects is constructed. Additional metadata that describes roles of columns in the fact and dimension metadata objects is generated. Also disclosed is a computer-readable medium for storing data for access by a program. A data

structure stored in the computer-readable medium includes data for use by the program. The data includes a cube model metadata object that includes a facts metadata object, one or more dimension metadata objects, and one or more join metadata objects that describe how one or more tables in the facts metadata object and one or more tables in the one or more dimension metadata objects are joined. The data also includes a cube metadata object that represents a subset of the cube model metadata object and comprises a view with columns for one or more measures of one of the facts metadata objects and one or more attributes of one or more of the dimension metadata objects and a document that describes roles of columns in the facts metadata object and the one or more dimension metadata objects.

Malloy teaches.

The Examiner has cited Malloy for ...

Malloy discloses the following.

"An OLAP multidimensional metadata system 100 includes multidimensional metadata software 120 (e.g., a stored procedure application programming interface (API)), a user interface 150, and multidimensional metadata objects 130. The multidimensional metadata software 120 is used to create, store, and access the multidimensional metadata objects 130. Optionally, a user interface 150 may be provided for a user or administrator to send commands to the multidimensional metadata software 120. A user may create, access, modify, or delete multidimensional metadata objects 130 by submitting commands via the user interface 150. The commands are received and processed by the multidimensional metadata software 120. For example, the multidimensional metadata software 120 may create and store multidimensional metadata objects 130.

In certain implementations, the OLAP multidimensional metadata system 100 provides an add-on feature for an RDBMS 110, such as DB2.RTM. Universal Database (referred to herein as DB2.RTM. UDB), that improves the ability of the RDBMS 110 to perform OLAP processing. The invention streamlines the deployment and management of OLAP solutions, and improves the performance of OLAP tools and applications.

In particular, the OLAP multidimensional metadata system 100 provides metadata objects. The new metadata objects are stored in, for example, a database catalog (e.g., the DB2.RTM. UDB catalog) that describes the dimensional model and OLAP constructs of existing relational data. The database catalog provides a single repository from which OLAP applications can capture multidimensional metadata. In certain implementations, the metadata objects may reside on a data store other than the database catalog or may reside across multiple data stores. With the information in the central repository, a database optimizer is able to use techniques specific to star schemas for optimizing the execution of queries.

With the metadata objects, the invention can optimize OLAP query performance by aggregating data in summary tables and creating indexes. The OLAP multidimensional metadata

system 100 also provides a metadata programming interface. In particular, the OLAP multidimensional metadata system 100 provides an SQL and extensible mark-up language (XML)-based application programming interface (API) for OLAP tools and application developers. XML is a text format defined by the World Wide Web Consortium (W3C) and further details on XML may be found at Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation 6 Oct. 2000, which is available at http://www.w3.org/TR/REC-xml. Through CLI, ODBC, or JDBC connections or by using, for example, embedded SQL to DB2.RTM. UDB, applications and tools can use a single stored procedure (i.e., an example of multidimensional metadata software 120) to create, modify, and retrieve metadata objects. In certain implementations, multiple stored procedures may provide the functionality for creating, modifying, and retrieving metadata objects.

OLAP multidimensional metadata system 100 metadata objects describe relational information as intelligent OLAP structures, but the multidimensional metadata objects provided by the invention are different from traditional OLAP objects. The metadata objects of the invention store metadata, meaning the metadata objects store information about the data in the base tables. Metadata objects describe where pertinent data is located and can also describe relationships within the base data. For example, a facts metadata object is a specific metadata object that stores information about related measures, attributes and joins, but does not include the data specifically from the base fact table.

Metadata provides a new perspective from which to understand data. Without metadata objects, a database catalog only knows about table and column names and cannot store information about the meanings of the tables and columns or how the tables and columns relate to each other. With metadata objects, this information may be stored.

Each metadata object completes a piece of the big picture showing what the relational data means. Some metadata objects act as a base to directly access relational data by aggregating data or directly corresponding to particular columns in relational tables. Other metadata objects describe relationships between the base metadata objects and link these base metadata objects together. Ultimately, all of the metadata objects can be grouped together by their relationships to each other, into a metadata object called a cube model. A cube model represents a particular grouping and configuration of relational tables. The purpose of a cube model is to describe OLAP structures to a given application or tool. Cube models tend to describe all cubes that different users might want for the data that are being analyzed. A cube model groups dimensions and facts, and offers the flexibility of multiple hierarchies for dimensions. A cube model conveys the structural information needed by query design tools and applications that generate complex queries on star schema databases".

(Malloy, paragraphs 59 – 65)

Given the context of Malloy, the two references actually teach away from one another. There can be no motivation to combine the teachings of Block which disclose a transformation of datastreams into various formats with the teachings of Malloy in which the various desired formats are already present. Applicants therefore submit that Block and Malloy or not permissibly combined, as there can be no motivation to combine their disparate teachings.

For these reasons, applicants respectfully submit that claim 1, as amended, is not anticipated or rendered obvious by the cited references, alone or in combination. Given that claims 18 and 23 include similar limitations and that claims 2 – 17, 19 – 22, and 24 -39 depend, directly or indirectly, from claims 1, 18, and 23, respectively, applicants reespectively submit that claims 2 – 39 are, likewise, is not anticipated or rendered obvious by Block or Malloy, alone or in combination.

In view of the foregoing, it is respectfully asserted that the claims are now in condition for allowance.

Conclusion

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Applicant respectfully requests that a timely Notice of Allowance be issued in this case. Please charge any additional required fee or credit any overpayment not otherwise paid or credited to our deposit account No. 50-1698.

Respectfully submitted,

THELEN REID & PRIEST, LLP

Dated: 10/12/02

Thomas Van Zandt
Reg. No. 43,219

Thelen Reid & Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
Tel. (408) 292-5800
Fax. (408) 287-8040